

Optimal decoding in fault-tolerant concatenated quantum error correction

Zachary W. E. Evans* and Ashley M. Stephens

Centre for Quantum Computer Technology, School of Physics

University of Melbourne, Victoria 3010, Australia

(Dated: February 26, 2009)

Here we present a new approach to concatenated quantum error correction in which additional classical processing is used with existing quantum codes and fault-tolerant circuits to more reliably correct errors. Using this new scheme we are able to guarantee correction of a number of discrete errors up to half of the distance of the concatenated code. We simulate the application of the scheme to the $[[4,1,2]]$ subsystem code with up to three levels of error correction.

PACS numbers: 03.67.Lx, 03.67.Pp

I. INTRODUCTION

Our expectation that a quantum computer will one day outperform a classical computer is in part founded upon quantum error correction, which will be necessary to mitigate environmental decoherence and errors that arise from imprecise quantum control. Various methods for error correction are known, including topological error correction [1, 2, 3], surface codes [1, 4, 5], and color codes [6]. The method we are concerned with in this paper involves concatenating small quantum codes that are analogous to classical codes such as the repetition code and the Hamming code.

Quantum codes such as the Bacon-Shor code [7, 8] and the Steane code [9] work by redundantly encoding the state of a single, physical qubit in the combined state of a few qubits, whereafter an error in the state of one or more of these qubits can be diagnosed and corrected using appropriately designed circuits that control the propagation of errors [10]. This reduces the error rate of an encoded quantum circuit relative to an unencoded quantum circuit by at least a factor proportional to reciprocal of the physical error rate, as two or more concurrent errors are required to cause failure. However, because there are more places for errors to occur than before, the encoded error rate will only be lower than the physical error rate if the physical error rate is below some threshold [11].

A single level of error correction can only reduce the encoded error rate so much. A greater reduction can be achieved by encoding a single physical qubit in the state of a few qubits, as before, and then encoding each of those qubits in the state of more qubits, and so on. This forms a concatenated quantum code, which allows, in theory, an arbitrarily long and large universal quantum computation to be undertaken reliably and efficiently. However, as both the number of qubits and the time required increase exponentially with the number of concatenation levels, it is likely that only a few levels will be practical. Therefore with finite resources the fault tolerant threshold should not be thought of as the critical point for

achieving arbitrary accuracy in quantum computing. To achieve a sufficiently low logical error rate with a practical amount of resources it is important to make each level as effective as possible at reducing the encoded error rate.

Typically, each level of a concatenated code operates independently of all other levels and implicitly assumes that errors on all qubits are equally likely. For a distance- d code that is concatenated l times, this method results in a concatenated code that can fail if there are $\lceil d/2 \rceil^l$ errors. However, as the total distance of the concatenated code is d^l , one might expect that a more sophisticated decoding method may exist which will never fail with fewer than $\lceil d^l/2 \rceil$ errors. Such a method is known for the case of quantum information being communicated over a noisy channel [12], but not for the case where errors can occur during error correction itself.

Recently we reported on a new method of concatenated error correction which makes use of classical information obtained during the first level of error correction to assist in the diagnosis of errors at the second level [13]. When applied to the $[[9,1,3]]$ subsystem code our method was shown to correct all cases with four physical errors whereas conventional error correction, where each level of error correction operates independently, can only guarantee success with up to three physical errors. Our method was inspired by a similar method that has been applied to an error-detection code with the result that the minimum number of errors that cause failure scales with increasing levels of concatenation as the Fibonacci sequence [14, 15].

Here we extend the method of Ref. [13] so that it is applicable to any number of levels of concatenation. By using information gathered by the process of error correction itself we determine the relative likelihood of failure at each circuit location and thus maximize the probability of correctly interpreting high-level error syndromes. Our method requires no additional quantum resources and increases the minimum number of errors that cause failure to half the total distance of the concatenated code. In the design of this method we assume a discrete error model in which the probability of error at each physical circuit location, p , is constant and independent of the other locations. We also assume that the quantum computer will be operating in the low- p limit, thus the exponent of the

*Electronic address: zevans@unimelb.edu.au

lowest order of p is sufficient to describe probabilities. It may be possible to improve the method by relaxing these assumptions.

II. METHOD

The key idea for our method is that the process of error correction provides information that can be used to estimate the probability that an encoded qubit has failed. For example, whenever error correction measures a non-zero syndrome we know that at least one error has occurred, and so our confidence in the encoded information should be decreased. Here we aim to use all of the information at our disposal to determine the most likely set of errors that could give rise to the measured syndrome so that we can maximize the probability of returning to the correct codeword state. We achieve this by tracking effect of each possible error through the error-correction circuitry, along with an associated confidence rating. Whenever a syndrome measurement is made, the results are compared to predicted syndrome measurements for possible sets of errors. We apply corrections to the data qubits based on the most likely set of errors that would give rise to the syndrome. Confidence ratings for the higher-level encoded qubits are calculated using the information from the syndrome measurements.

A. Flag raising

For each location in the error-correction circuit we determine what effect a discrete X and Z error occurring at that location would have on the data and ancilla at the end of syndrome extraction. Taking into account that some errors are equivalent to other errors, we assign a *flag* to each unique error. In other words, there is a flag for each possible effect on the data and ancilla that can be caused by a single error in the circuit. Each flag represents the possibility of a single error. The probability of any given flag representing an actual error is described by the *weight* of the flag such that $\text{Pr}(\text{error}) = \mathcal{O}(p^{\text{weight}})$. As the possibility of error at each location is independent of the other locations, the probability of multiple flags representing actual errors is $\mathcal{O}(p^{\sum \text{weights}})$, where the sum is over the set of flags being considered. Error correction and flags in the X and Z bases are treated separately.

At the physical level, each location in the error-correction circuit is assigned a confidence of one, indicating that physical-level locations all fail with probability p . At the end of each error-correction cycle (in each basis), after measuring the syndrome and applying the appropriate correction, a confidence rating is calculated for the encoded qubit based on the syndrome and flags. This is the logical confidence rating of the higher-level circuit location, the calculation of which will be described in the following section. Initially, all flags can be thought of as

having infinite confidence (as there cannot be any errors before any operations have been done). As each circuit location is executed, the weight of the flag corresponding to an error at that location is updated to be the minimum of its current value and the confidence reported at that location. Detection of errors at lower levels will lower our confidence of success at particular circuit locations all the way up to the highest level of error correction.

Two-qubit gates are special in that errors may be shared on both qubits. For example, errors occurring before a transversal CNOT may be copied and errors occurring during a CNOT may affect both the control and the target qubits. Therefore, the confidence ratings reported by error correction immediately following a two-qubit gate are used to update the flag which corresponds to a correlated error affecting both qubits involved in the gate. This is in addition to updating both of the flags corresponding to the uncorrelated errors after gate. The flag weight corresponding to the two-qubit error is taken to be the maximum of the two single-qubit confidence ratings as opposed to the sum, which would correspond to the two uncorrelated errors. The possibility of a correlated error is thus represented by its own flag.

Any error on a data qubit will eventually have its effect transferred down to the ancilla. The location of the error determines which ancillary qubit will see its effect. An error which affects a data qubit only after it has interacted with the ancilla in its current error-correction cycle can not be detected until the following error-correction cycle. Therefore, flags that represent errors that effect only data qubits should be put aside until the current error-correction cycle is complete, after which they will represent errors that will be detected in the following error-correction cycle. These flags can be called *transitive flags*. Following error correction, each transitive flag is converted into its associated non-transitive flag for use in the next error-correction cycle. (See Section III for an example.)

Each logical qubit has its own set of flag weights. When two logical qubits interact (during a logical operation) certain errors within one logical qubit may propagate onto the other logical qubit. Let us call the logical qubit from which errors may be copied the *source* and the logical qubit to which errors may be copied the *target*. Flags representing the possibility of errors on the source are used to update the weights of flags on the target. For each possible error that may be copied from the source, the corresponding flag weight on the target is taken to be the minimum of its current weight and the weight of the flag on the source. Although copying an error from one logical qubit to another would actually result in a correlated error affecting both qubits, we treat each error and its associated probability as if they were independent so that error correction can be performed on each logical qubit separately. The possibility of correlated errors is reintroduced at the logical level where the weight is taken to be the maximum of the weights of the individual logical qubits.

B. Flag matching, correction, confidence reporting

At the end of each error-correction cycle the ancillary qubits are measured to obtain the syndrome. Each possible syndrome is consistent with a finite number of sets of errors. Each of these sets of errors has a corresponding set of flags. We call each set that matches the measured syndrome a *flag match*. The most likely cause of the syndrome is the set of errors that is represented by the flag match with the lowest weight, where the weight of a flag match is the sum of the weights of the flags in the set. With the lowest weight flag match identified, we apply corrections to whichever data qubits would be affected by the set of errors represented by the match.

Alternative flag matches will imply different data corrections. In particular, there will always be a flag match which implies a complement set of corrections to that of the lowest weight match – that is, the corrections of both matches combine to form a logical operator. If we correct based on some match but the true set of errors are represented by its complement match then the result is a logical failure. To determine the logical confidence rating that will be used by error correction at the level above, we consider the probability that our choice of correction will result in a logical error,

$$\Pr(\text{fail}|\mathcal{S} \cap \text{flags}) = \frac{\Pr(\text{fail} \cap \mathcal{S}|\text{flags})}{\Pr(\mathcal{S}|\text{flags})}. \quad (1)$$

$\Pr(\text{fail}|\mathcal{S} \cap \text{flags})$ is the probability of a logical failure given the current syndrome and set of flags. $\Pr(\mathcal{S}|\text{flags})$ is the probability of the current syndrome occurring given the current set of flags, regardless of success or failure. $\Pr(\text{fail} \cap \mathcal{S}|\text{flags})$ is the probability of the current syndrome occurring and error correction resulting in a logical error. In the low- p limit these probabilities are dominated by the leading-order terms and the coefficients are not important. Since our method applies corrections based on the most likely cause of the syndrome, to leading order $\Pr(\text{SS}) = \Pr(\text{success} \cap \text{SS})$. This implies

$$\frac{\Pr(\text{fail} \cap \mathcal{S}|\text{flags})}{\Pr(\text{success} \cap \mathcal{S}|\text{flags})} \approx p^{y-x} \quad (2)$$

where x is the weight of the lowest weight flag match and y is the weight of the complement match. Therefore,

$$C_L = y - x \quad (3)$$

is the logical confidence rating calculated during this error-correction cycle.

The logical confidence rating represents the probability of a logical error, $\mathcal{O}(p^{C_L})$. However, there is also a possibility that the corrections applied will result in a state that is outside of the code space, so that it is neither the correct state nor the state affected by a logical error. Just as we used the complement match to determine the probability of a logical error we can determine the probability of individual errors on each data qubit by

considering the other flag matches. At the end of each error-correction cycle the confidence of each data qubit is updated to be the minimum of the confidence of its associated transitive flag and a confidence determined during the error correction of the encoded qubit, where this confidence is calculated in an analogous way to the logical confidence.

In large codes there will also be probabilities of correlated sets of data errors that we are able to derive from the flag matches. Let E represent a particular set of a data errors after an error-correction cycle. Then

$$\Pr(E|\mathcal{S} \cap \text{flags}) = \frac{\Pr(E \cap \mathcal{S}|\text{flags})}{\Pr(\mathcal{S}|\text{flags})}. \quad (4)$$

The weight of the flag corresponding to this set of data errors should, therefore, be updated to be the minimum of its current weight and the sum of each set of flags that match this outcome, taking into account any corrections that have been applied, minus the weight of the minimum weight flag match. (See Section III for an example.)

By the careful consideration of every possible cause of every syndrome we always have accurate confidence ratings for every qubit at every level. We use this information to maximize the probability of success by always correcting based on the most likely set of errors.

III. APPLICATION

Here we apply our method to the $[[4,1,2]]$ subsystem code [7, 8]. The stabilizer generators of this code are

$$X_1 X_2 X_3 X_4, \quad Z_1 Z_2 Z_3 Z_4, \quad (5)$$

where X_i and Z_i respectively represent the Pauli operators σ_X and σ_Z applied to the i^{th} qubit. Tensor products and identity operators are implicitly present. The gauge group of this code is generated by the operators

$$\begin{aligned} & X_1 X_2, \\ & X_3 X_4, \\ & Z_1 Z_3, \\ & Z_2 Z_4. \end{aligned} \quad (6)$$

The logical X and Z operators are

$$\begin{aligned} X_L &= X_1 X_4, \\ Z_L &= Z_1 Z_4, \end{aligned} \quad (7)$$

or equivalently any operators that can be generated by combining Eq. 7 with the gauge operators of Eq. 6.

The syndrome-extraction circuit which we have chosen to use is shown in Fig. 1 [16], along with the flags that will be used in our message-passing method. The circuit performs operator measurements of the gauge operators, two at a time. Since the stabilizers are products of the

gauge operators, the combined parity of each pair of ancilla measurements is effectively the result of an operator measurement of a stabilizer. Individually, each gauge operator measurement tells us nothing about the syndrome.

Since the logical operators of the $[[4,1,2]]$ code are pairwise products of single-qubit operators, the code distance is equal to two. This means that it is only an error-detection code. The syndrome of the code consists of a single bit in each basis. A syndrome of zero indicates that no error was detected; one indicates that an error was detected but provides no information on which qubits were affected by the error. When the syndrome is measured to be one, in the absence of any additional information, we must simply guess which data qubit was affected by the error. Due to the gauge group of the code, each data qubit has a partner qubit in each basis for which errors are equivalent. For example, an X error on the first qubit is the same as an X error on the second qubit up to the application of the X_1X_2 gauge operator. Such an error can, therefore, be corrected by the application of an X operator on either the first or second qubit. To simplify the analysis of errors, we refer to errors by which gauge operator that they are a part of, rather than by which data qubit they affect. An X error on the first or second qubit is said to be an error on the *first gauge* in the X basis, a Z error on the second or fourth qubit is said to be an error on the *second gauge* in the Z basis, and so on. When there is an error on one gauge and the syndrome is misinterpreted so that the correction operation is applied to the other gauge, the result is always a logical error.

In order to reliably correct errors using the $[[4,1,2]]$ code, extra information is required to distinguish between errors affecting the first and second gauges. One method for obtaining this information is to pass classical messages describing the relative probabilities of errors from lower levels of error correction to higher levels. Previously, it was shown that a message passing scheme such as this could increase the minimum number of errors that cause failure to a number that scales according to the Fibonacci series with increasing levels of concatenation [14, 15]. In contrast, the scaling achieved by our method is given by 2^{l-1} where l is the number of levels of error correction.

It is worth noting that we have chosen to use $[[4,1,2]]$ code for several reasons. Since it is a small code we can simulate more levels of error correction than if we had used a higher distance error-correction code. As our method involves an analysis of combinations of errors that can occur at every circuit location, it is easier to understand and test the operation of the method on a code that requires only small circuits with relatively few locations. Finally, a similar code forms the basis of the post-selection scheme of Knill, where ancillary states are prepared and verified before they are used to enact error correction by teleportation. The advantages of this scheme include a high threshold and fast logical gates, though unfortunately these come at a high resource cost. Our method might be used to improve the performance

Flag	Effects
$AG1$	first gauge and ancilla
$AG2$	second gauge and ancilla
A	ancilla only
$G1$	first gauge only (transitive)
$G2$	second gauge only (transitive)

TABLE I: Names assigned to the flags of the $[[4,1,2]]$ code. There are five flags for each basis. *First gauge* refers to an error on either of the two qubits involved in the first gauge operator of the appropriate basis in Eq. 6, similarly for *second gauge*. *Ancilla* refers to an error which flips the parity of the two gauge operator measurements – that is, an error which changes the outcome of the syndrome measurement. Flags $G1$ and $G2$ become $AG1$ and $AG2$ respectively at the end of the error-correction cycle as they will be seen on the ancilla of the following error-correction cycle.

and overhead of the scheme.

A. Technique

For the $[[4,1,2]]$ code, using the circuits we have chosen, there are five unique effects that can result from a single error. Therefore there are five flags that we are required to track for each logical qubit at each level of error correction. These flags are defined in Table I. Note that two of the flags are transitive. The set of error locations belonging to each flag are shown in Fig. 1.

In our scheme the errors represented by the flag match with the lowest weight are identified as being the most likely cause of the observed syndrome. Corrections are applied to any data qubits marked by the flag match. For example, if the X basis syndrome is one, and the weights of the X flags are $AG1 = 4$, $AG2 = 0$, $A = 2$, then the flag match with the lowest weight is $AG2$ and so a correction should be applied to the second gauge – that is, an X correction should be applied to either qubit three or qubit four. Table II shows a list of all possible flag matches along with the various confidence ratings that would result from each of them being acted on.

B. Simulation and results

Our simulations are of a CNOT extended rectangle with concatenated error correction, including all physical locations. The message-passing scheme is simulated along side the concatenated error correction circuit so that it operates in the same way as it would in a real quantum computer. The logical CNOT extended rectangle is defined to succeed only if both logical qubits would, after decoding, produce the correct measurement results in both the X and Z bases. The simulations use the method outlined in [13], which involves keeping track of discrete Pauli errors as they propagate through a circuit. We

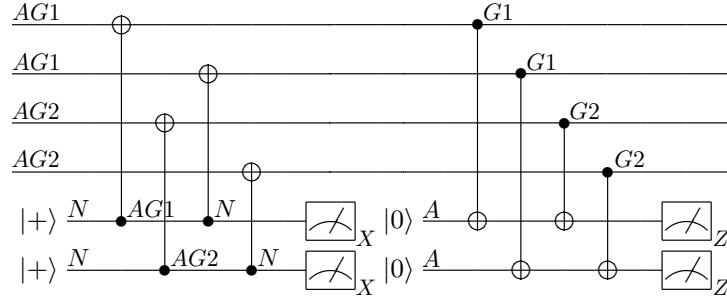


FIG. 1: Syndrome-extraction circuit for the $[[4,1,2]]$ code, with X flag mapping shown. The error syndromes in the X and Z bases are obtained by performing operator measurements of the gauge operators in Eq. 6. The parity of the two X gauge operator measurements gives the Z error syndrome and vice versa. The letters on the circuit mark out regions associated with each of the X flags. The confidence rating of each circuit location is associated with the flag that is labeled to the left of that location. (See Tab. I for the flag labels). For example, an X error that occurs in the region marked $AG1$ will affect one of the qubits involved the first X gauge operator and the effect will be seen on the ancilla used to perform the operator measurement. An X error that occurs in the region marked A will flip the parity of the ancilla but will not affect the data at all. An X error that occurs in the region marked N will have no effect on the data or the observed syndrome. The flag regions shown here are for X flags only but the Z flag map is similar.

Syndrome	Flag match	Correction	C_{G1}	C_{G2}	C_L
0	None	None	$AG1 + A$	$AG2 + A$	$AG1 + AG2$
	$AG1 + A$			N/A	
	$AG2 + A$			N/A	
	$AG1 + AG2$			N/A	
1	$AG1$	$G1$	$A - AG1$	$AG2 + A$	$AG2 - AG1$
	$AG2$	$G2$	$AG1 + A$	$A - AG2$	$AG1 - AG2$
	A	None	$AG1 - A$	$AG2 - A$	$AG1 + AG2$
	$AG1 + AG2 + A$			N/A	

TABLE II: Table of all flag matches for each of the two possible syndromes. Corrections and flag updates are based on the flag match with the lowest weight from the list in the second column. The correction and flag update rules associated with each flag match are shown in the last four columns. C_{G1} and C_{G2} are confidence ratings for the transitive flags $G1$ and $G2$ at the current level of error correction. C_L is the logical confidence rating for the next-highest level of error correction.

assume a stochastic error model, where memory, initialization, readout, single-qubit gate, and two-qubit gate errors are all equally likely to occur. Errors are randomly selected one- and two-qubit discrete Pauli errors. Our simulator uses the SIMD-oriented Mersenne Twister pseudo random number generator [17]. We use the error-correction circuit shown in Fig. 1, which requires two ancilla qubits per logical qubit for a total block size of six qubits.

We are primarily interested in the logical failure rate when the physical error rate, p , is low. Directly simulating the circuit in the low- p limit requires too many runs to obtain statistically significant results. Instead, we use the expansion of logical failure rate with respect to the physical error rate,

$$p_L = \sum_{i=0}^N r_i \binom{N}{i} p^i (1-p)^{N-i}, \quad (8)$$

where N is the number of physical locations in the error-correction circuit, p is the physical error rate, and r_i is the

probability of logical failure after error correction given exactly i errors. To estimate values of r_i we simulate the circuit with exactly i errors placed randomly, repeating many times for each i . We approximate p_L by truncating the series above $i = 29$. This approximation breaks down for values of p for which the probability of having more than 29 errors becomes significant.

The failure rate of the logical CNOT as a function of physical error rate is plotted in Fig. 2 for one, two, and three levels of error-correction. In the low- p region, the gradient of each line on the log-log plot signifies the minimum number of errors that are required for the circuit to fail: one, two, and four errors for one, two, and three levels of error correction respectively.

IV. FURTHER WORK

As described, our scheme assumes that each physical location is equally likely to fail. However, this need not be the case. Relative probabilities of failure for individ-

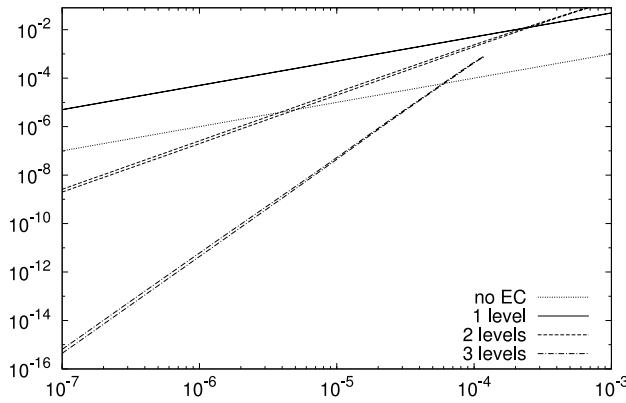


FIG. 2: Failure rate of CNOT extended rectangle as a function of physical error rate for one, two, and three levels of error correction with the $[[4,1,2]]$ code and our message-passing method. 2σ of statistical error above and below the mean is shown for each set of data. The minimum number of discrete errors that can cause failure is one, two, and four for one, two, and three levels of error correction respectively. This can be inferred by the gradients of the lines in the low- p region. The asymptotic threshold appears to be around 2×10^{-4} . Since the first level of the code is unable to reliably correct any errors, the failure rate for one level of error correction is greater than the physical failure rate.

ual physical locations are naturally represented in our scheme by the confidences at the physical level. That is, rather than setting the confidence of all physical locations to one, we can choose to assign different confidence ratings to each type of operation or each individual physical device. These confidence ratings could be based on data obtained by characterization of the gates. Alternatively, data collected during error correction could be

fed back to dynamically update physical confidence ratings in a similar way to how our method updates logical confidence ratings.

Our message-passing method could be applied to a post-selection based scheme instead of the Fibonacci scheme. There are several ways this could be done, since we are free to choose the confidence threshold which determines whether ancillary states are accepted or rejected. To only accept states that have the maximum confidence would reduce the scheme to ordinary post-selection, as a state would be rejected if any errors are detected at any level regardless of whether or not these errors are correctable at higher levels. Accepting all states that have a confidence greater than one would result in a high acceptance rate, but would not be optimal with respect to logical fidelity. In any case, the confidence of the states that are accepted could be used in higher-level error correction following post selection. There may also be some advantage to using the logical confidence ratings outputted by the highest level of error correction in some quantum algorithms.

Our method can be applied to higher distance Bacon-Shor codes and perhaps even to other CSS stabilizer codes. It would be interesting to investigate the pros and cons of using different codes with our method and also to rigorously establish the specific requirements on codes and circuits that enable us to achieve optimality with respect to the minimum number of errors that cause the code to fail.

Acknowledgements: We acknowledge financial support from the Australian Research Council, the US National Security Agency, and the Army Research Office under contract number W911NF-04-1-0290.

[1] A. Kitaev. *Ann. Phys.* **303**, 2 (2003).
[2] C. Nayak, S. H. Simon, A. Stern, M. Freedman, S. Das Sarma. arXiv:0707.1889 (2007).
[3] R. Raussendorf, J. Harrington, and K. Goyal. *New J. Phys.* **9**, 199 (2007).
[4] S. Bravyi and A. Kitaev. arXiv:quant-ph/9811052 (1998).
[5] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. *J. Math. Phys.* **43**, 4452 (2002).
[6] H. Bombin and M. A. Martin-Delgado. *Phys. Rev. Lett.* **97**, 180501 (2006).
[7] D. Bacon. *Phys. Rev. A* **73**, 012340 (2006).
[8] D. Bacon and A. Casaccino. *Proc. Ann. Alerton Conf.* **44**, arXiv:quant-ph/0610088 (2006).
[9] A. M. Steane. *Phys. Rev. Lett.* **78**, 2252 (1997).
[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2000).
[11] P. Aliferis, D. Gottesman, and J. Preskill. *Quant. Inf. Comp.* **6**, 97 (2006).
[12] D. Poulin. *Phys. Rev. A* **74**, 052333 (2006).
[13] Z. W. E. Evans and A. M. Stephens. *Phys. Rev. A* **78**, 062317 (2008).
[14] E. Knill. *Nature* **434**, 39 (2005).
[15] P. Aliferis and J. Preskill. *Phys. Rev. A* **79**, 012332 (2009).
[16] P. Aliferis and A. W. Cross. *Phys. Rev. Lett.* **98**, 220502 (2007).
[17] M. Saito and M. Matsumoto. *Monte Carlo and Quasi-Monte Carlo Methods* **2**, 607 (2006).